

UNITED STATES PATENT APPLICATION
FOR
**METHOD AND APPARATUS FOR THE DIGITAL CREATION OF FUR
EFFECTS**

INVENTOR:

ARMIN WALTER BRUDERLIN

PREPARED BY:

BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN LLP
12400 WILSHIRE BOULEVARD
SEVENTH FLOOR
LOS ANGELES, CA 90025-1026

(408) 720-8598

"Express Mail" mailing label number EL3710060336US

Date of Deposit 1/16/02

I hereby certify that I am causing this paper or fee to be deposited with the United States Postal Service "Express Mail Post Office to Addressee" service under 37 CFR 1.10 on the date indicated above and is addressed to the Commissioner of Patents and Trademarks, Washington, D.C. 20231

Jenny Vinter
(Typed or printed name of person mailing paper or fee)

Jenny Vinter
(Signature of person mailing paper or fee) Date

METHOD AND APPARATUS FOR THE DIGITAL CREATION OF FUR EFFECTS

BACKGROUND OF THE INVENTION

1. FIELD OF THE INVENTION

5

The invention relates to the digital creation of fur. More particularly, the present invention relates to the digital creation of realistic close-up and distant looks of fur coats on animal models.

2. ART BACKGROUND

10 One of the many challenges in modeling, animating and rendering believable mammals in computer graphics has been to produce realistic-looking fur. A real fur coat is made up of hundreds of thousands of individual, cylindrical hairs covering the skin, and fulfills vital functions such as protection against cold and predators. Between animals as well as across the body of
15 individual animals, the look and structure of these hairs vary greatly with respect to length, thickness, shape, color, orientation and under/overcoat composition. In addition, fur is not static, but moves and breaks up as a result of the motion of the underlying skin and muscles, and also due to external influences, such as wind and water.

20 Some prior computer graphics techniques used for fur creation have achieved convincing looks of smooth fur; however, these techniques do not take into account that real fur often breaks up at certain areas of the body, such as around the neck. In addition, the prior methods do not account for hairs of wet fur clump together resulting in a significantly different appearance compared to
25 dry fur. Also, the process of simulating hair as it is getting increasingly wet when sprinkled on by water has not yet been addressed.

40052066-041620

SUMMARY OF THE INVENTION

The system and method of the present invention provides a flexible technique for the digital representation and generation of realistic fur coats on geometric models of surfaces, such as animals. In one embodiment, an innovative technique for placement, adjustment and combing of fur on surfaces is provided. In one embodiment, the continuity of fur across surface patch boundaries is maintained. In addition, in one embodiment, an innovative method to simulate wet fur is provided. In this method static clumping and animated clumping may be applied to regions on the surfaces. In one embodiment, a method for the symmetric and one-sided breaking of hairs along fur-tracks on surfaces is provided. The above processes can be iteratively applied in order to generate layers of fur, such as an undercoat and an overcoat.

BRIEF DESCRIPTION OF THE DRAWINGS

The objects, features and advantages of the present invention will be apparent from the following detailed description in which:

5 **Figures 1a and 1b** are simplified block diagrams of embodiments of systems that operate in accordance with the teachings of the present invention.

Figure 2 is a flow diagram of one embodiment of a process for the generation of fur in accordance with the teachings of the present invention.

10 **Figure 3a** is an illustration of a set of parametric surfaces defining the skin of a three-dimensional animal model.

Figure 3b is a simplified flow diagram illustrating one embodiment of static and animated combing processes.

Figure 3c and 3d are examples that illustrate one embodiment of the combing processes.

15 **Figure 4** is a flow chart illustrating one embodiment of a process for adjusting control hairs for removing visual discontinuities at surface boundaries.

Figure 5 is a flow chart illustrating one embodiment of a process for placing hairs.

20 **Figure 6** illustrates one example of subpatches defined on the surface.

Figure 7a illustrates an example of control vertices of one control hair.

Figure 7b illustrates an example for calculating control hair weights.

25 **Figure 7c** illustrates an example an interpolation process used to calculate orientations of final hairs in accordance with the teachings of one embodiment of the present invention.

Figure 7d is a simplified flow diagram of one embodiment for the calculation of the orientation of final hairs.

Figure 8 is a flow chart illustrating one embodiment of a process to perform static clumping.

Figure 9 illustrates examples of different clump-percent and clump-rate values.

Figure 10a shows a rendered frame of a combed fur coat and Figures 10b, 10c and 10d show snapshots of one embodiment of an animated dry-to-wet fur sequence.

Figure 11 is a flow chart illustrating one embodiment of a process for animated area clumping.

Figure 12a is a flow chart illustrating one embodiment of a process for hair breaking.

Figure 12b illustrates examples of symmetric and one-sided breaking of hairs.

Figures 12c, 12d, 12e, and 12f illustrate examples of breaking effects.

Figures 13a, 13b, and 13c illustrate the visual effects of undercoat and overcoat.

Figure 14 is a flow chart illustrating one embodiment of a shading process.

1000000000000000

DETAILED DESCRIPTION

The method and apparatus provides an innovative technique for the digital generation of fur on surfaces, such as on a computer generated animal.

Figure 1a is a simplified block diagram of one embodiment that operates in accordance with the teachings of the present invention. Computer system 10 includes central processing unit (CPU) 15, memory 25, Input/Output 20, which may be coupled to a storage device such as a disk drive or other device. The system may also include a keyboard 40 or other user input device as well as a display 35 that may be used to display user interface and the final rendering of the fur in accordance with the teachings of the present invention.

In one embodiment, memory 25 stores instructions which when executed by the CPU 15 performs the processes described herein. Alternately, the instructions may be received via storage 30 or other input such as user input 40. The processes described herein may be executed via software by a system such as system 10 or hardware, or a combination of both hardware and software.

An alternate embodiment is illustrated in Figure 1b. Input is received by surface definition module 50 that defines a surface which, as will be explained below, defines surfaces and control hairs of the object to be rendered. Module 55 adjusts the control hairs to provide such functionality as combing and seamless hairs across surface boundaries. The interpolation module 60 interpolates across the surfaces using the control hairs. Hair clumping and breaking module 65 enhances the realistic visualization of the object by providing for clumping and breaking of hairs. Rendering module 70 renders the hairs and provides shading, black lighting and shadowing effects to the hairs, and module 75 displays the final output of the object with the hair surfaces

Figure 2 is a flow diagram of the steps involved in generating fur coat in accordance with the teachings of the present invention. At step 200, the geometry of the surfaces that contain the hair is defined. In one embodiment, a three-dimensional geometry may be used to model the skin, for example an

animal skin, on which the fur coat is later generated. As illustrated in **Figure 3a**, the geometry is usually defined as a connected set of parametric surfaces often referred to as surface patches. The patches can be generated a number of ways known to one skilled in the art. In one embodiment, NURBS surface 5 patches are used.

Referring back to **Figure 2**, at step 210, control hairs are placed onto these surface patches, whereby each control hair is modeled as a parametric curve, e.g., a NURBS curve, defined by a user-specified number of control vertices. As will be discussed below, a global density value for the hair is given by the user 10 to determine the number of actual hairs and their positions on the surface patches. Each hair also has a number of attributes such as length, width, waviness, opacity, and by default points in the direction of the surface normal at its position on the surface.

In the present embodiment, a number of operations are performed on 15 control hairs and the final hairs are generated based upon the control hairs and other information. However, it should be realized that these steps, such as combing and the like described herein may be performed on the final hairs instead of the control hairs.

A number of different approaches may be taken to generate the control 20 hairs. One simple algorithm places equally spaced x hairs in the u direction and y hairs in the v direction (where x and y are specified by the user) of each NURBS patch. Alternately, the x and y hairs are placed equally by arc-length. This will result in a more uniform distribution across the patch. However, it does not achieve a balanced distribution of control hairs across patches of 25 different sizes; x and y hairs are placed on all selected patches, whether large or small. Therefore, in an alternate embodiment, the generation of control hairs takes into account the area of a NURBS patch to determine x and y for each patch. In one embodiment, the user specifies z hairs per unit area. In addition in one embodiment, control hairs can also be placed individually or along 30 curves-on surfaces for finer control. For example, extra control hairs along the

sharp rim of the ears of an animal can be generated to ensure proper alignment of the fur eventually generated.

Referring back to **Figure 2**, once the control hairs are generated, step 210, the control hairs are adjusted at the surface boundaries. Since control hairs are placed in each surface patch, control hairs that lie on the boundary of a surface patch might not line up with control hairs on a neighboring surface patch; this can lead to visible discontinuities of the hairs along the surface boundaries. To address the potential problem, the control hairs on surface boundaries are adjusted. One embodiment of the process for adjusting the control hairs is illustrated by the flow chart of **Figure 4**.

At step 400, seams are constructed between adjacent surfaces. Each seam identifies adjacent surfaces along a corresponding boundary (for example, an entire edge, T-junctions, or corners) of a surface patch. At step 405, for each surface patch, the boundaries are traversed, step 410. Each control hair is examined, step 412. At step 415, if a boundary hair is found, at step 420, the neighboring patches, as identified by a corresponding seam, are checked to see if there is a corresponding hair on the neighboring patch. In one embodiment, a hair is corresponding if it is within a small predetermined distance from the boundary hair. The distance may be specified in parametric u, v, or absolute space. In one embodiment, the predetermined distance may be a relatively small distance such that the hairs visually appear co-located.

If there is a corresponding control hair, the boundary hair and the corresponding hair are aligned, step 425, by modifying the location and orientation of one or both of the control hairs to a common location and orientation, respectively. In one embodiment, the corresponding hair of an adjacent surface patch is snapped to the location of the boundary hair along the boundary. In one embodiment, if the neighboring surface patch does not have a corresponding hair, then one is inserted and aligned on the neighboring patch, step 445. The process continues for each boundary hair along each

boundary in each surface patch, steps 430, 435 and 440 until all boundary hairs are aligned.

Referring back to **Figure 2**, after step 215, in one embodiment, the control hairs have been placed on the surfaces defining an animal or other object model 5 and the control hairs point along a surface normal at their positions on the surface.

At step 220, the hairs are combed to achieve desired, groomed, dry fur looks. A number of different combing processes may be used. In the present embodiment, however, static and animated combing processes are applied to 10 the control hairs. The combination of static and animated combing provides a low computation cost but effective visual effect. In alternate embodiments, static or animated combing alone may be used and generate beneficial visual results. The combing process may be used on the same control hairs for different shots, to provide, for example, a groomed look versus a slightly messy 15 look of the fur.

One embodiment will be described with reference to **Figure 3b**. In one embodiment, static combing is applied if the hairs do not "actively" move during an animation of the object, e.g., the animal. It should be noted that since each hair is expressed in a local coordinate system defined by the surface 20 normal, du and dv at the root of the hair, statically combed hairs will "passively" move as the underlying surface is deformed or animated. Combing is achieved by specification of combing direction curves, degree of bending and curvature of the hair, as well as with a fall-off for each combing direction curve.

At step 325, one or more combing direction curves are created. These 25 curves indicate the direction applicable control hairs are to be combed. An example is illustrated by **Figures 3c and 3d**. **Figure 3c** shows a number of uncombed control hairs. **Figure 3d** illustrates an exemplary combing direction curve 365 and its direction. The combed hairs as also illustrated in **Figure 3d**.

Figure 3d illustrates one combing direction curve. However, it is 30 common to implement a number of different curves, each curve corresponding

to a different area of the surface. Thus, at step 330, for each curve, one or more control hairs are assigned such that the assigned hairs are combed in accordance with the corresponding combing direction curve.

In addition, at step 335, for each curve, bend, curvature and fallout parameters are defined. The bend parameter defines how close the control hair is to the surface. The curvature parameter indicates the shape of the hair. For example, a curvature value of zero may indicate the hair is to be straight up and a maximum value (e.g., 1) may indicate the hair is bent on a tight arc from root to tip.

The fallout value indicates a region beyond which the combing direction curve decreases its influence the further away the control hair is from the curve. In some embodiments, the fallout region is specified to cover relatively large areas so all control hairs are equally affected and fallout does not occur. In other embodiments it is desirable to decrease the combing effect, the further the distance between the control hair and the combing direction curve.

At step 340, each control hair is processed according to the combing direction curve it is assigned to, and the bend, curvature and fallout parameters. The result of the process is illustrated by Figure 3d in which the hairs are combed in the direction of the combing direction curve 365. In addition, the hairs are curved according to the bend and curvature parameters defined. In the present example the fallout parameter defines the entire surface such all hairs are equally affected and fallout is not present.

As noted above, animated combining may also be applied, step 345. Key framing, known in the art, is used to interpolate between combing changes specified at certain frames to proving smooth transitions between changes. Thus for example, bend curvature and fallout parameters may be specified to change at certain frames. The key framing process execution then transitions during the frames between the specified frame changes. This technique can be used to simulate a variety of conditions which affect the look of the hair, such as

wind. Thus, the hairs can be animated by key framing the parameters and executing the combing calculations at each frame during playback.

The combing process may also include a simple hair/surface collision model process in which hairs that intersect with the underlying surface due to 5 the combing process are pushed back up to be above the surface. Hairs may be rotated to intersect the underlying surface due to, for example, by setting the bend parameter to a large value.

The process includes an iterative algorithm that determines hair/surface intersections. For example, the process performs a line segment intersection 10 check of successive control vertices of a curve (e.g., the NURBS curve) defining a control hair with the surface. If a control vertex c goes below the surface, the hair is rotated back towards the surface normal from the previous non-intersecting vertex just enough for c to clear the surface. The amount of rotation is large enough to cause the hair to rotate back up above the surface by 15 a small amount specified by the application. Thus the vertices of the vector affected by the combing are rotated back towards the surface normal so that the vector is above the surface.

In an alternate embodiment, the combing may be animated by turning each control vertex of a control hair into a particle, and applying dynamic 20 effects like gravity and external forces. Software, such as Maya, available by Alias | Wavefront, a division of Silicon Graphics, Inc., Toronto Canada, may be used to perform this function.

Once the control hairs are identified and processed (e.g., adjusted, combed), the final hairs for each patch are generated from the control hairs, step 25 223, **Figure 2**. As noted above, in one embodiment the control hairs are first adjusted along surface boundaries. In an alternate embodiment, combing may be applied to control hairs alone or in conjunction with application of the surface boundary adjustment process.

One exemplary process for the placement of hairs on patches is 30 illustrated by the flow chart of **Figure 5**. In this embodiment, final hairs are

generated from control hairs in two set steps. First, the static hair features are calculated, e.g., the placement (the u , v position) of the final hairs. This step may be performed once. The second set of steps may be performed for each frame in an animation and provide frame dependent hair features.

5 At step 505, subpatches are identified on a surface of the object on which the hairs are to be generated. **Figure 6** illustrates one embodiment for placing the final hairs on the surfaces defining the underlying skin. In the present embodiment, the root positions for each final hair are determined in terms of (u, v) parametric values of the surface. They are computed from an overall
10 (global to all surfaces) density input value d_{max} (number of hairs per square unit area), and a set of normalized local density values (values range from 0 to 1; default value may be 1) per surface patch, arranged at a variable resolution grid of equally spaced points on the surface (for example, 128x128 points).

In one embodiment, the process attempts to make the number of hairs
15 independent of the resolution of this grid and independent of the surface patch size to provide seamless density across surfaces of different scale. For purposes of discussion, it is assumed that the specified input density value (d_{max}) is 10 hairs/unit square area, and the local density values are arranged at equally spaced points on the surface as shown in **Figure 6** (e.g., 0.4, 0.5, 0.6, 0.6 hairs,
20 respectively). These points define the subpatches of the surface to be processed, step 505, **Figure 5**. As these equally spaced points are traversed, step 510, the area in (u, v) space between the neighboring points is approximated by the area defined by two polygons, more particularly, triangles (a_1 and a_2), and the number of hairs/square unit area for each triangle hair unit is averaged
25 from the values at its vertices step 520. In one embodiment, this is determined according to the following: $\text{HairUnit} = d_{max} * V_{avg}$, where d_{max} represents the specified input density value and V_{avg} represents the average local density value for each triangle determined from values at its vertices. For the example defined, this results in $10 * (0.4 + 0.5 + 0.6) / 3 = 5$ and $10 * (0.4 + 0.6 + 0.6) / 3 = 5.333$
30 hairs/square unit area for the top left and bottom right triangle, respectively.

At step 525, the total number of hairs to place on the current subpatch is determined from the actual approximated area of the subpatch (a1 and a2) and the number of hairs per unit area. In one embodiment, the total number of hairs per unit area is determined according to the following: HairTotal =

- 5 $A * \text{HairUnit}$, where A represents the actual approximated area of the subpatch. For example, if a value of 0.4 applies to area a1 and 0.3 applies to area a2, as assumed for purposes of discussion, $0.4 * 5 + 0.3 * 5.333 = 3.5999$ is the total number of hairs to place in subpatch defined by $(ui, vi), (ui, vi+1), (ui+1, vi+1), (ui+1, vi)$.

At step 530, the final hairs are placed. Since it is preferable not to place fractional hairs, either 3 or 4 hairs are placed depending on whether a uniformly generated random number in $[0,1]$ is bigger or smaller than the fractional part (0.5999). The 3 or 4 control hairs are randomly placed in $u [ui, ui+1]$ and randomly in $v [vi, vi+1]$. The process then proceeds back to step 515 to the subpatch defined by the next four equally spaced points.

15 Each final hair contains a number of control vertices. The root position (first control vertex) of each control hair is specified in terms of a (u,v) value of the underlying surface. The remaining control vertices of each hair are defined in a known local coordinate system with origins specified at the hair root position, and axes in the direction of the surface normal, du, dv. In one embodiment, each hair is oriented along the surface normal and the coordinates of the control vertices are generated by subdividing the length of each hair into n-1 equal parts, where n is the number of control vertices/hair. One example is illustrated in **Figure 7a**, where a hair 725 is defined on surface 730 with n=4. The root is vertex 720 and the remaining vertices are 705, 710 and 715.

25 Once the root position is calculated the enclosing control hairs (in one embodiment three) for each final hair are determined. In one embodiment, a 2-dimensional Delaunay triangulation (know in the art and therefore not further discussed herein) is constructed of the (u,v) positions of the control hairs for each surface patch. This triangulation was chosen because it creates "well-proportioned" triangles, by minimizing the circumcircle and maximizing the

minimal angles of the triangles. Once the Delaunay triangulation is constructed, it is determined which triangle each final hair falls into. The indices of the three control hairs which form the particular triangle are assigned to the hair that falls into that triangle.

5 The weights (w_1, w_2, w_3) which each of the three control hairs (c_1, c_2, c_3) has on the final hair (h) are then calculated. This may be done using barycentric coordinates (know in the art and not further discussed here), and is illustrated in **Figure 7b**, where "A" represents the area of triangle 726 (c_1, c_2, c_3). These weights are used for interpolating the final hairs from the control hairs as
10 explained below.

The above information of each final hair (i.e., the (u, v) position, the 3 enclosing control hairs, and the weights of each control hair) may be generated only once for an object in animation. This information is referred to herein as the static information. In contrast, the calculation of the orientation of each final
15 hair may be done at each frame of an animation. This orientation is determined from the orientation of the control hairs and their corresponding weights by an interpolation process as explained with reference to **Figures 7c and 7d**.

For each final hair (h) at step 756, the corresponding three control hairs (c_1, c_2, c_3) are converted into a surface patch space (in one embodiment, the
20 patch coordinate system) utilized. At step 758, control hair vectors, (e.g., v_{11}, v_{12}, v_{13}), between control vertices are calculated (e.g., 782, 783). A variety of techniques may be used to calculate the control hair vectors; in one embodiment, the vectors are equally distributed between control vertices. The control hair vectors are then normalized (e.g., $nv_{11}, nv_{12}, nv_{13}$). At step 760
25 the corresponding control hair vectors of the three control hairs are interpolated and multiplied by the three determined weights that were calculated for the final hair. In one embodiment, for example, one control hair vector is determined according to the following equation:

$$iv_1 = nv_{11} * w_1 + nv_{12} * w_2 + nv_{13} * w_3;$$

wherein iv1 represents the interpolated, weighted control hair vector representation of the final hair, nv11, nv21 and nv31 represent normalized control vectors and w1, w2 and w3 represent the corresponding weights for the normalized control vectors. At step 762, the resulting vectors are scaled to a
5 final hair length (siv1, siv2, siv3). Once the scaled vectors are determined, at step 764 the control vertices (786, 788, 790, 792) of the final hair are calculated from the scaled vectors.

As illustrated in **Figure 2**, steps 225, 230, 235 and 245 may be optional and may not be necessary to produce a groomed, dry fur coat. Steps 225, and
10 230 are applied for the generation of wet fur. Clumping of hairs can occur when the fur gets wet due to the surface tension or cohesion of water. The effect is that the tips of neighboring hairs (a bunch of hairs) tend to gravitate towards the same point, creating a kind of cone-shaped "super-hair", or circular clump. As will be described below, step 225 is executed for static area clumping
15 that generates hair clumps in fixed predefined areas. Step 230 is executed for animated area clumping , that is, when clumping areas move on the model, for example, to simulate spouts of water or raindrops hitting the fur and making it increasingly wet. In both cases, parameters which can be animated are provided to achieve various degrees of dry-to-wet fur looks. Step 235 is
20 applied to generate dry fur clumping or breaking.

According to a particular application all or some of the steps 225, 230 and 235 may be executed. In addition, the steps 225, 230 and 235 may be prioritized according to an application such that a hair adjusted in a higher priority step is not adjusted in the other steps. Alternately, the effects may be cumulative or
25 selectively cumulative depending upon a particular application.

At step 225 static area clumping is performed. One embodiment of this process is described with reference to **Figure 8**. For purposes of discussion, the center hair of each clump is referred to as the clump-center hair, and all the other member hairs of that clump, which are attracted to this clump center hair,
30 are referred to herein as clump hairs.

In one embodiment there are four clumping input parameters: clump-density, clump-size, clump-percent and clump-rate. Similar to the hair-density parameter, clump-density specifies how many clumps should be generated per square area. The process described herein translates the clump density into an
5 actual number of clumps defined by clump-center hairs, the number of clump center hairs depending on the size of each surface patch. As a result, some of the existing hairs are turned into clump-center hairs.

Clump-size defines the area of a clump. In one embodiment, the clump size is defined in world space, a space that a user typically references with
10 respect to a size of an object. In one embodiment, clump density takes priority over clump size, such that if there are many clumps and most of them overlap, the clump size cannot be maintained, since a clump hair can be a member of only one clump. If both clump density and size are small, many hairs between the clumps will not be clumped.
15

Referring to **Figure 8**, to determine clump membership of each final hair (i.e., what clump each hair belongs to, if any), the clump of the specified clump-size is converted into u-radius and v-radius components in parametric surface space at each clump-center hair location, step 800. Each hair is evaluated at steps 805, 810 to determine whether it falls within the u, v radius components of
20 a corresponding clump-center hair. If the hair is not within the u, v radius components, the hair is not a clump hair, step 815 and the process continues, step 830, with the next hair. If the hair is within the u, v radius components, at step 820 the clump-center hair's index is referenced with the hair. In addition, a clump rate and clump percent is assigned, step 825.

25 A number of variations are contemplated. A clump-size noise parameter may be introduced to produce random variations in the size of the clumps. Feature (texture) maps for a clump-size can be created and specified by the user, one per surface patch, to provide local control of the radii used at steps 805, 810. In this embodiment, the global clump-size input parameter is
30 multiplied for a particular clump (clump center hair) at (u,v) on a surface patch

with the corresponding normalized (s,t) value in the clump-size feature map for that surface. Also, a static clump-area feature map can be provided to limit clumping to specified areas on surface patches rather than the whole model.

In one embodiment a clump-percent and clump-rate value is assigned to each clump hair (step 825). In one embodiment, the values for both range between [0,1], and are used subsequently to reorient clump hairs, step 835, as described below.

Clump-percent specifies the degree of clumping for a clump hair. For example, a value of zero indicates that the hair is not clumped at all, i.e., it is like a "dry" hair. A value of one indicates that the hair is fully attracted to its clump-center hair, i.e., the tip of the hair (its distant control vertex) is in the same location as the tip of the clump-center hair.

Clump-rate defines how tightly a clump hair clumps with its corresponding clump-center hair. For example, a value of zero indicates that the clump hair is linearly increasingly attracted to its clump-center hair, from the root to the tip. A clump-rate value closer to one indicates that the hair's control vertices closer to the root are proportionally more attracted to corresponding clump-center hair vertices than those closer to the tip, which results in tighter clumps. Examples of different values for clump-percent and clump-rate are given in **Figure 9**.

At step 835, the control vertices (except the root vertex) of each clump hair are reoriented towards corresponding clump-center hair vertices from the clump hair's dry, combed position determined at steps 200, 210, 215, 220, and 223.

In one embodiment, this process is performed at each frame. In one embodiment, the default value for number of control vertices (CVs) is 3 (4 minus the root vertex), and the index for the current control vertex *i* ranges from 1-3. In one embodiment, the reorientation is determined as follows:

clumpHairCV[i] = clumpHairCV[i]+delta* (clumpCenterHairCV[i] -clumpHairCV[i])

delta = clumpPercent * (fract + clumpRate* (1 - fract)); where fract = i /
numberOfCVs; clumpHairCV[i] represents a clump hair vertex;
clumpCenterHairCV[i] represents a corresponding clump center hair vertex; i
5 represents an index to a current control vertex; numberofCVs represents the
number of control vertices of a clump hair; clumpPercent represents clump-
percent; and clumpRate represents the clump-rate.

Both clump-percent and clump-rate parameters can be locally controlled
via feature maps similar to the feature maps described above with respect to
10 clump-size. Both values can also be animated or changed over time to provide
continuous control for dry-to-wet-to-dry fur looks. This is illustrated by
Figures 10a, 10b, 10c and 10d which illustrate four frames from an animated
clump-percent and clump-rate sequence. In the image of **Figure 10a** the clump-
percent and clump-rate are both zero and may represent dry, combed hair. In
15 the image of **Figure 10b**, clump-percent is 0.7 and clump-rate is 0, which results
in a slightly wet look. In the image of **Figure 10c**, clump-percent is 1.0 and
clump-rate is 0.3, which results in a wet look. In the image of **Figure 10d**,
clump-percent and clump-rate are both 1.0, which produces a very wet look.

Animated area clumping is desirable to simulate spouts of water or
20 raindrops hitting the fur and making it increasingly wet. At step 230, **Figure 2**,
animated clumping is performed. In one embodiment, the animated clumping
areas are defined in an animation system.

One embodiment of the process is described with reference to **Figure 11**.
In one embodiment, clumping areas are defined by particles hitting surface
25 patches. Other embodiments may use alternate techniques for generating
animated clumping areas. At step 1100 a global static area clumping process is
performed on all hairs. This step identifies clumping regions and
corresponding clump center hairs and clump hairs. As explained below this
information is used in the animated clumping process. In one embodiment, the

global static area clumping used is that described above for static area clumping.

At step 1102, one or more emitters that generate the particles are defined. The use of emitters to generate particles is known in the art and will not be discussed in detail herein. In one embodiment, the emitters define the rate generated and spread of the particles across a surface.

At step 1105, at each frame for each particle generated that hits the surface, the surface patch the particle hits is identified, step 1110. In one embodiment, particles generated in prior frames are carried through subsequent frames such that the particles are cumulative.

For each particle that hits a surface patch, including those particles generated in prior frames, a circular animated clumping area is created, step 1115, on the patch at that (u,v) location, with clump-percent, clump-rate, and animated clumping area radius determined by a creation expression executed at the frame where the particle hits the surface so that when a particle hits the surface at that time (i.e., at the frame), the clump-percent may be set to zero and the radius may be defined to a specified value perhaps adjusted by a random noise value. Thus, the expression may be defined to provide the desired "wetness" effect.

The radius of the circular clumping area defined is converted into a corresponding u-radius and v-radius similar to the clump size discussed above. Runtime expressions executed at each frame define clump-percent and clump-rate, thus determining how quickly and how much the fur "gets" wet. For example, one runtime expression may be: MIN(FrameNumber * 0.1, 1) such that as the frame number increases, the hair appears increasingly wet.

Each clump center hair of a clump (determined at step 1100) is then evaluated to determine if it falls within the animated clumping area, step 1120. To determine whether a clump falls within an animated clumping area, at each frame it is checked as to whether the (u,v) distance between the clump-center hair of the clump and the center of the animated clumping area is within the

(u,v) radius parameters of the animated clumping area. For clumps that are located in overlapping animated clumping areas, the values for clump-percent and clump-rate are added resulting in the generation of wetter fur.

If the clump center hair is within the animated clumping area, step 1125, 5 the corresponding clump is flagged with an animated clumping flag such that the clump hairs are subsequently reoriented to reflect the animated clumping effect. Alternately, each clump hair of the clump may have an animated clumping flag which is set if the corresponding clump center hair is determined to be within an animated clumping area. In addition, an animated clump-rate 10 value and an animated clump-percent value are assigned to the clump hairs that are identified to be within an animated clumping area in accordance with a runtime expression. In one embodiment, the values for clump-percent and clump-rate for each clump within an animated clumping area are replaced with the corresponding values for the animated clumping area at each frame. As 15 animated clumping areas may be much bigger than a clump, an animated clumping area may contain several individual clumps. Each clump is evaluated, step 1140, for each particle, step 1145.

It should be noted that animated clumping areas could straddle surface patch boundaries. For example, the center of an animated clumping area may 20 be located on one surface patch, but the area may be located on one or more other patches. Since the animated clumping areas are typically defined and therefore associated with the surface which contains the center of the animated clumping area, i.e., the position where the particle hit, portions of a clumping area straddling neighboring patches may be overlooked. This could lead to 25 discontinuities in clumping of the final fur.

In one embodiment, this potential problem is addressed. Whenever a new particle hits a surface and the (u,v) radii exceed the boundaries of that surface; an additional (u,v) center and (u,v) radii is generated for the animated clumping areas affecting neighboring patches. Thus, for example, if the 30 clumping area covers portions of two neighboring patches, a corresponding

(u,v) center and radii are generated for each neighboring patch to provide additional animated clumping areas for evaluation at steps 1120-1140

At step 1150, for each frame, the clump hairs of clumps that are within the animated clumping areas are reoriented. Thus, clump hairs are selectively adjusted if they are within an animated clumping area. In one embodiment, clumping is restricted to the animated clumping areas at each frame, so that final hairs of clumps outside the animated clumping areas are normally produced as "dry" hairs.

At step 1155, if more frames are to be processed, the process continues again at step 1105. Thus the animated clumping process is performed across multiple frames to provide animated effects.

Referring back to **Figure 2**, step 235 may be applied to generate the effect of hair breaking or dry fur clumping by breaking up the groomed fur coat along certain lines (fur tracks or break line) on the underlying skin (surfaces).

As described below, this process may include two kinds of hair breaking: symmetric and one-sided. In symmetric breaking, hairs on both sides of a fur-track "break" towards that track, whereas in one-sided breaking, hairs on one side of the track break away from the track.

In one embodiment, fur tracks are specified as curves on surfaces in an animation system. Each track has a radius, break-percent and break-rate for symmetric and one-sided breaking, and an additional break-vector for one sided breaking. The final information generated is output into breaking files that are subsequently accessed to reorient the affected hairs.

One embodiment of the hair breaking technique is illustrated by **Figure 12a**. At step 1200 the fur tracks are defined. The fur tracks may be defined similar to clumps by defining a (u,v) break radii. At step 1205 the break line hairs (hairs which lie on or are very close to the fur-track curve defined by the curve defined for the fur track) are computed. Using the break line hairs and break radii, at steps 1215, 1220, each hair is evaluated to determine whether the hair lies within the (u,v) break radii on both sides of the break line hairs in case

of symmetric breaking, or to one side specified by the break vector (the break vector side) in case of one- sided breaking. For each hair within the space specified by the radii, referred to herein as a break hair, the corresponding break line hair (hair on the fur track) is then determined as the one closest to it.

- 5 The hairs are labeled as break line hairs, break hairs with indices to their corresponding break line hairs, or normal hairs that do not reside within the areas specified by the break.

It should be noted that for instances of one-sided breaking, each break hair is now reoriented "away" from its corresponding break line hair in the direction of the break-vector, rather than "towards" the break line hair.

10 Examples of symmetric and one-sided breaking are shown in **Figure 12b**.

The break hairs are reoriented with respect to their corresponding break line hairs, step 237. For symmetric breaking, this process is analogous to the process performed for clump hairs discussed earlier. However, for break hairs, 15 the break-percent and break-rate values are used in place of the clump-percent and clump-rate used for clump hairs. For one-sided breaking, break hairs are repelled, as opposed to attracted to the break-line hairs according to the break-percent and break-rate parameters.

The breaking effect is illustrated by **Figures 12c, 12d 12e and 12f**. **Figure 20 12c** illustrates an object 1250 having break line hairs 1252, 1254. **Figure 12d** shows the resultant effect on the object for symmetric breaking. **Figure 12e and 12f** illustrate one-sided breaking along break line hairs 1256-1270.

At step 245, **Figure 2**, a decision is made as to whether multiple passes of the process are to be performed. The coat of most furred animals is composed 25 of a fuzzier, thinner shorter layer of hairs called undercoat, plus an overcoat of longer and thicker hairs. Step 245 illustrates the capability to perform a two-(or multiple)-pass process, whereby steps 210, 215, 220 and 223 (and optionally 225, 230, and 235) are executed more than once, producing a different set of hairs at each pass. These sets or layers are then processed and combined at render-time 30 (step 250). The effects can be seen by reference to **Figure 13a, 13b and 13c**.

Figure 13a is a representation of an undercoat generated in accordance with the teaching of the present invention. **Figure 13b** represents a representation of the overcoat and **Figure 13c** represents the combined image consisting of the undercoat and overcoat.

5 As illustrated by step 250, the clumped hairs represented by their control vertices are rendered into a series of two-dimensional images to create lifelike dry and wet hair looks. In one embodiment, the process functions to project a three-dimensional hair geometry onto a two-dimensional image plane from the perspective of a particular point of view.

10 In order to render large amounts of hair quickly and efficiently, the geometric model of each hair may be kept simple. As explained above, a hair is represented by a parametric curve having a determined number of control vertices (in one embodiment, the default is four).

15 In one embodiment, the process employs known rendering technology to produce the hairs described by the corresponding control vertices. In an alternate embodiment customized modules are added to realistically "shade" the hairs. This may be accomplished by assigning an intensity of color at each point along or on a hair, wherein points along a hair may be defined as the pixels which compose the hair.

20 During the rendering of the hairs, a width is added for each hair to transform it into a narrow ribbon that is always oriented towards the camera or view point. The shading process properly shades these ribbon primitives to more realistically present them as thin hairs.

One embodiment of the shading process is set forth in the flow chart of
25 **Figure 14**. At step 1400, each hair is processed. At step 1405, for each hair, the surface normal at the base of the hair is mixed with the normal vector at the current point on the hair in order to obtain a shading normal at the current point on the hair. In one embodiment, the hairs are rendered as a series of points or pixels on a display. Thus, the current point is one of the pixels
30 representing a hair.

The shading process may be applied at multiple points along the hair. In one embodiment, the amount with which each of these vectors contributes to the mix is based on the angle between the tangent vector at the current point on the hair, and the surface normal vector at the base of the hair. The smaller this
5 angle, the more the surface normal contributes to the shading normal.

At step 1410 the intensity of the hair at the current point on the hair is determined using the shading normal at that point. In one embodiment, a Lambertian model is used to calculate these intensities. Using this approach provides the benefit of allowing the user to light the underlying skin surface
10 and receive predictable results when fur is added. This approach also accounts for shading differences between individual hairs, and differences in shading along the length of each hair.

In order to obtain realistic shadows on the fur coat, shadow maps are used. The use of shadow maps is known in the art and will not be discussed
15 further herein. However, incorporating the hair into the shadow maps may generate several unwanted side effects. One problem is that of dark streaking on brightly lit fur because of the fur self-shadowing. Dark streaks look wrong on brightly lit fur because normally light bounces off the skin and hair to prevent dark shadows on brightly lit fur.

20 In order to minimize the dark streaking effects, in one embodiment, the hairs for the shadow maps are shortened based on certain criteria, step 1415. For example, the length and density of the hair may dictate the percentage to shorten the hair. By selectively shortening hairs for the shadow maps, the hair self-shadowing effect is minimized while still producing a broken up shadow
25 on the terminator lines for lights falling on the fur.

Back lighting is achieved in a similar fashion using a shadow map for each light located behind the furred object, and again shortening the hair on the basis of density and length in the shadow map render process. In one embodiment, a lighting model for hairs also allows each light to control its
30 diffuse fall-off angles. Thus, lights directly behind the furred object can wrap

around the object. Using these lighting controls together with shadow maps reasonable back lighting effects are achieved.

In one embodiment, the shading for clumped hairs is modified. In one embodiment, two aspects of the hair shading may be modified. First, the amount of specular on the fur is increased. Second, clumping is accounted for in the shading model. Geometrically, as explained earlier, fur is modeled in clumps to simulate what actually happens when fur gets wet. In the shading model, for each hair and for each light, the side of the clump the hair is on with respect to the light's position is determined, and the hair is either darkened or brightened based on the side the hair is on. Thus, hairs on the side of a clump facing the light are brighter than hairs on a clump facing away from the light.

The invention has been described in conjunction with the preferred embodiment. It is evident that numerous alternatives, modifications, variations and uses will be apparent to those skilled in the art in light of the foregoing description.